Information-aware Lyapunov-based MPC in a feedback-feedforward control strategy for autonomous robots

Olga Napolitano[1], Daniele Fontanelli[2], Lucia Pallottino[1] and Paolo Salaris[1]

***Abstract*—This paper proposes a feedback-feedforward control scheme that combines the benefits of an online active sensing control strategy (the feedforward control component) to maximize the information needed for correctly executing the desired task, with a Lyapunov-based control strategy (the feedback control component) that guarantees an asymptotic convergence towards the task itself. To quantify the amount of the collected information along the planned trajectories, the smallest eigenvalue of the Constructability Gramian is adopted as a metric and optimized, for generating the feedforward control component, within a Lyapunov-based Model Predictive Control framework (LMPC). The latter indeed allows to systematically handle the closed-loop stability and robustness properties of a Lyapunov-based nonlinear control law, and, at the same time, to reduce the estimation uncertainty and, thus, increase the task execution performance. To show the effectiveness of our method, we consider three case studies where a unicycle equipped with suitable onboard sensors has to perform three classical tasks in mobile robotics: path following, point-to-point motion, and trajectory tracking.**

***Index Terms*—Optimization and Optimal Control; Sensor-based Control; Motion and Path Planning**

I. INTRODUCTION

ACTION selection is a crucial decision process for humans, and depends on the state of both their body and the environment that, analogously to robots, sensors cannot provide directly. Because signals in humans sensory and motor systems are affected by variability and/or noise, the brain dedicates a lot of effort to efficiently combine collected information (i.e., sensor inputs) and prior knowledge (i.e., the knowledge base from life memories) [1]. In particular, humans appear to adopt a strategy that is not a pure feedback, but includes also a feedforward active sensing control component to reduce the detrimental effects of uncertainty [2] and hence increase the probability of task success w.r.t. passive sensing [3].

Manuscript received: September, 9, 2021; Revised December, 15, 2021; Accepted January, 26, 2022.

This paper was recommended for publication by Editor E. Manchand upon evaluation of the Associate Editor and Reviewers' comments.

This work has received funding from European Union's Horizon 2020 Research an Innovation Program under Grant Agreement No. 101017274 (DARKO) and partially by the Italian Ministry of Education and Research (MIUR) in the framework of the CrossLab project (Departments of Excellence).

[1] Research Center "E. Piaggio" and Dip. of Information Engineering, University of Pisa, Italy. paolo.salaris,lucia.pallottino@unipi.it olga.napolitano@phd.unipi.it

[2] Department of Industrial Engineering, University of Trento, Italy. daniele.fontanelli@unitn.it

Digital Object Identifier (DOI): see top of this page.

In robotics, action, and motion planning [4] are typically used to accomplish a given task (e.g., reaching a particular configuration) with stability guarantees (e.g., Lyapunov stability theory), and/or optimizing a cost of interest (e.g., control effort), under different constraints (e.g., on limited Field-of-View sensors). However, as for humans, the successful generation and execution of a motion plan substantially depends on the accuracy of the reconstructed surroundings and (internal) state trajectories that, in a real scenario, are not assumed directly measurable by on-board sensors but estimated. Due to non-linearities, the quality of the sensory information strongly depends on the actions chosen to perform the task, as for humans. Similarly, an interesting coupling also exists in robotics between perception and action: generation of a motion/action plan should find a balance between efficient/stable task execution and improved estimation.

This paper hence proposes a feedback-feedforward strategy for a robotic system where the feedforward component aims at maximizing the information collected by the onboard sensors (for correctly accomplishing a desired task) by using an online active sensing control strategy [5], while the feedback component guarantees an asymptotic convergence, in the Lyapunov sense, towards the desired state/task (e.g., reaching a desired posture).

Active sensing control strategies are widely used in robotics to reduce estimation uncertainty. In [6] authors surveyed the fundamental components of robotic active learning systems, while [7] uses active perception to improve the quality of domain randomization-based pose estimation with neural networks applied to 2D images. A task-oriented active sensing scheme that minimizes the uncertainty in future task-related actions is proposed in [8], whereas [9] proposes a yaw-based trajectory control algorithm that jointly optimizes aggressiveness and feature co-visibility for state estimation improvement. In [10], instead, a perception-aware model predictive control framework for quadrotors has been proposed to maximize the visibility of a point of interest and minimize its velocity in the image plane. The authors of this paper have already proposed active sensing control strategies [11], taking also into account measurements noise [12], intermittent measurements [13] as well as in combination with shared control [14]. However, we never consider the coupling of an active sensing control strategies with a stabilizing feedback control law for better accomplishing a task of interest. We will show here that control inputs computed on the state estimates generate an uncertainty on the time derivative of the candidate Lyapunov

function proportional to the state estimation uncertainty, which is directly related to the active sensing choices. To quantify the amount of the collected information (and hence of the uncertainty) along the planned trajectories, the *Constructability Gramian* (CG), quantifying the level of constructibility of the current/future state, is used as the guiding metric [11], [15].

Ensuring Lyapunov stability with uncertainty is a challenging problem that can be tackled using authority-sharing paradigms [16]. Instead, in this paper, the effective combination of the feedback/feedforward components with uncertainties is pursued adopting an iterative, finite-horizon optimal control approach within the Lyapunov-based Model Predictive Control (LMPC) technique [17, Chapter 2]. LMPC technique is classically used to account for control input bounds in combination with, e.g., measurements delays [18], data losses [19], trajectory tracking performance requirements [20]. [21] presents a real-time solution for a unified Nonlinear MPC and Control Lyapunov Function controller with limited computational resources. Finally, [22] presents an MPC approach for a class of nonlinear systems with unbounded uncertainties guaranteeing stochastic stability. However, none of the previous publications deals with the intimate and fruitful connection between feedback and feedforward components, which are here presented using simulations on a unicycle vehicle engaged in three classical tasks in robotics: path following, trajectory tracking and point-to-point motion.

II. INFORMATION-AWARE LYAPUNOV-BASED MPC

The components of the proposed feedback-feedforward control scheme are detailed in this section and depicted in Fig. 1: the Lyapunov-based feedback control law steers the robot to the task accomplishment, while the feedforward component maximizes the sensor information through active sensing.

A. THE FEEDBACK COMPONENT

Let us consider a time-invariant, input affine nonlinear system

$$\dot{\boldsymbol{q}}(t) = \boldsymbol{f}(\boldsymbol{q}(t)) + \boldsymbol{g}(\boldsymbol{q}(t))\boldsymbol{u}(t) \tag{1}$$

where $\boldsymbol{q}(t) \in \mathbb{R}^n$ is the state of the system, $\boldsymbol{u}(t) \in \mathbb{R}^m$ its control input and $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are the drift vector and the control vector field, respectively. Let us then consider a positive definite candidate of Lyapunov $V(\boldsymbol{q}(t))$, with $V(0) = 0$ and $\boldsymbol{q}(t) = 0$ the desired equilibrium. The Lyapunov-based Control Law $\boldsymbol{u}(t) = \boldsymbol{u}_{fb}(\boldsymbol{q}(t))$ (LCL in Fig. 1) that makes $\boldsymbol{q}(t) = 0$ asymptotically stable is derived by imposing

$$\dot{V}(\boldsymbol{q}(t)) = L_f V(\boldsymbol{q}(t)) + L_g V(\boldsymbol{q}(t))\boldsymbol{u}_{fb}(\boldsymbol{q}(t))) \leq 0, \tag{2}$$

with L the Lie derivative, and then applying, if needed, the LaSalle-Krasowski principle.

The above control design implicitly assumes that the state of the nonlinear system $\boldsymbol{q}(t)$ is completely known. However, in a real scenario, the state of the system is usually unknown, and only an estimate $\hat{\boldsymbol{q}}(t)$ is made available by an observer (e.g., an EKF) which exploits sensory data. As a consequence, the control inputs $\hat{\boldsymbol{u}}_{fb}(\hat{\boldsymbol{q}}(t))$ are computed on the state estimates, which of course are affected by uncertainties. For this reason, the time derivative of V becomes,

$$\dot{V}(\boldsymbol{q}(t), \hat{\boldsymbol{q}}(t)) = L_f V(\boldsymbol{q}(t)) + L_g V(\boldsymbol{q}(t))\hat{\boldsymbol{u}}_{fb}(\hat{\boldsymbol{q}}(t)). \tag{3}$$

By adding and subtracting $L_g V(\boldsymbol{q}(t))\boldsymbol{u}_{fb}(q(t))$ in (3),

$$\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}}) = -L_g V(q)(\boldsymbol{u}_{fb}(q) - \boldsymbol{u}_{fb}(\hat{\boldsymbol{q}}) + \dot{V}(q). \tag{4}$$

Since an EKF will be adopted as an observer for the state estimation, we can assume that up to the first order

$$\hat{\boldsymbol{q}} = \boldsymbol{q} + \boldsymbol{\varepsilon}_q, \text{ with } \mathrm{E}\{\boldsymbol{\varepsilon}_q\} = 0 \tag{5}$$

where $\mathrm{E}\{\cdot\}$ is the expectation operator and, assuming that $\boldsymbol{\varepsilon}_q$ is the estimation error and $\boldsymbol{P}$ its covariance matrix returned by the EKF, we have $\boldsymbol{P} = \mathrm{E}\left\{\boldsymbol{\varepsilon}_q \boldsymbol{\varepsilon}_q^T\right\}$. It then follows that the effects of the uncertainties on the control law are described by the Taylor expansion around $\boldsymbol{\varepsilon}_q = \mathrm{E}\{\boldsymbol{\varepsilon}_q\} = 0$

$$\begin{aligned} \hat{\boldsymbol{u}}_{fb}(\hat{\boldsymbol{q}}) &= \boldsymbol{u}_{fb}(\boldsymbol{q}) + \boldsymbol{\delta}_{\boldsymbol{u}}(\boldsymbol{\varepsilon}_q) = \\ &= \boldsymbol{u}_{fb}(\boldsymbol{q}) + \left.\frac{\partial \hat{\boldsymbol{u}}_{fb}(\hat{\boldsymbol{q}})}{\partial \boldsymbol{\varepsilon}_q}\right|_{\boldsymbol{\varepsilon}_q = 0} \boldsymbol{\varepsilon}_q + O(\boldsymbol{\varepsilon}_q^2), \end{aligned} \tag{6}$$

where $O(\boldsymbol{\varepsilon}_q)$ are all the terms of order greater or equal to two. We can now determine the first two moments of the random variable (4) assuming the following first order Taylor approximation

$$\boldsymbol{\delta}_{\boldsymbol{u}}(\boldsymbol{\varepsilon}_q) \approx \left.\frac{\partial \hat{\boldsymbol{u}}_{fb}(\hat{\boldsymbol{q}})}{\partial \boldsymbol{\varepsilon}_q}\right|_{\boldsymbol{\varepsilon}_q = 0} \boldsymbol{\varepsilon}_q = \boldsymbol{D}\boldsymbol{\varepsilon}_q, \tag{7}$$

which yields (up to the first order approximation)

$$\begin{aligned} &\mathrm{E}\left\{\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}})\right\} = \dot{V}(\boldsymbol{q}), \\ &\mathrm{E}\left\{\left(\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}}) - \mathrm{E}\left\{\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}})\right\}\right)^2\right\} = \\ &= (-L_g V(\boldsymbol{q}))\boldsymbol{D}\boldsymbol{P}\boldsymbol{D}^T(-L_g V(\boldsymbol{q}))^T. \end{aligned} \tag{8}$$

From (8), it is clear that the stability of the equilibrium depends on the state estimation uncertainty and its propagation through the control inputs. Therefore, the use of an appropriate feedforward control component that steers the robots along the most informative trajectories to reduce the estimation uncertainty, has a positive impact on the task execution stability.

B. THE FEEDFORWARD COMPONENT

The MPC synthesizes a constrained optimal control sequence over a finite prediction horizon capable of handling control and state constraints as well as other optimization targets. The cost function to be minimized is typically a quadratic cost function involving penalties on the system states and control actions. MPC usually optimizes over a family of piecewise constant inputs with a fixed sampling time. Once the optimization problem is solved, only the first step of the control sequence is applied in a receding horizon fashion. Closed-loop stability is a common optimization constraint, e.g., inherited from a Lyapunov-based approach. In such a case, the MPC can be modified so that Lyapunov-based stability is enforced by design in the so called Lyapunov-based Model Predictive Control (LMPC) [17, Chapter 2] approach,

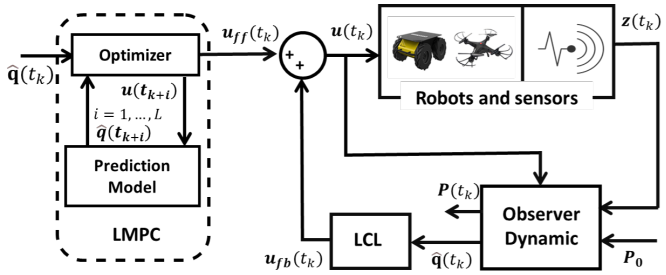


Fig. 1. Feedback-feedforward control scheme that determines at run-time the feedforward control action that maximizes the information collected through sensors along the future path taking into account the feedback control action that guarantees the asymptotic convergence toward the desired task.

whose classical formulation applies to affine control systems, as described as in the following Problem 1.

Problem 1 (LMPC) *Given the prediction horizon L, the control input $\boldsymbol{u}(t)$, the predicted trajectory of the nominal system $\tilde{\boldsymbol{q}}$ subjected to $\boldsymbol{u}(t)$ and with initial state $\boldsymbol{q}(t_k)$ at time t_k, find, $\forall t \in [t_k,\, t_{k+L}]$, the optimal sequence*

$$\boldsymbol{u}^* = \min_{\boldsymbol{u}\in S(\Delta)} J_{[t_k,\, t_{k+L}]}(\tilde{\boldsymbol{q}}(t), \boldsymbol{u}(t)) \tag{9}$$

s.t.

$$1)\ \dot{\tilde{\boldsymbol{q}}}(t) = \boldsymbol{f}(\tilde{\boldsymbol{q}}(t)) + \boldsymbol{g}(\tilde{\boldsymbol{q}}(t))\boldsymbol{u}(t) \tag{10}$$

$$2)\ \tilde{\boldsymbol{q}}(t_k) = \boldsymbol{q}(t_k) \tag{11}$$

$$3)\ \underline{\boldsymbol{u}} \le \boldsymbol{u}(t) \le \bar{\boldsymbol{u}} \tag{12}$$

$$4)\ L_g V(\boldsymbol{q}(t_k))\boldsymbol{u}(t_k) \le L_g V(\boldsymbol{q}(t_k))\boldsymbol{u}_{fb}(\boldsymbol{q}(t_k)) \tag{13}$$

where $S(\Delta)$ is the family of piece-wise constant functions with sampling period Δ, $J_{[t_k,\, t_{k+L}]}(\tilde{\boldsymbol{q}}(t), \boldsymbol{u}(t))$ is the cost index to be minimized, (10) *is the nominal model of the system, which is used to predict the state evolution starting from the initial state* (11)*, and* (12) *are the control constraints. Finally,* (13) *is the Lyapunov stability constraint guaranteeing that the time derivative of $V(\boldsymbol{q})$, computed at time t_k with $\boldsymbol{u}(t_k)$ the first step of the control strategy , is smaller than or equal to the value obtained if $\boldsymbol{u}_{fb}(\boldsymbol{q})$ would have been in a sample-and-hold fashion.*

With respect to the ideal Problem 1 we have to deal with two major issues. First, only an estimate of the state $\hat{\boldsymbol{q}}(t_k)$ is available at time t_k, hence it should be used in place of the actual initial state $\boldsymbol{q}(t_k) = \tilde{\boldsymbol{q}}(t_k)$, implying that $\boldsymbol{u}(t_k)$ turns to $\hat{\boldsymbol{u}}(t_k)$. Second, the proposed feedback-feedforward control scheme assumes that the first step of the MPC synthesized control law fed to the system is given by $\hat{\boldsymbol{u}}(t_k) = \hat{\boldsymbol{u}}_{ff}(t_k) + \hat{\boldsymbol{u}}_{fb}(t_k)$ (see Fig. 1). By substituting in (13), the stability constraint boils down to

$$L_g V(\boldsymbol{q}(t_k))(\hat{\boldsymbol{u}}_{ff}(t_k)+\hat{\boldsymbol{u}}_{fb}(t_k)) \le L_g V(\boldsymbol{q}(t_k))\hat{\boldsymbol{u}}_{fb}(t_k) \tag{14}$$

Furthermore, in the standard formulation of the LMPC, the cost function (9) usually is a weighted sum of the state and the control input [17, Chapter 2]. Instead, in this active sensing setting, the cost function introduced in [11] is adopted, which is subsumed in the following.

Let us consider a general nonlinear system

$$\begin{aligned} \dot{\boldsymbol{q}}(t) &= \boldsymbol{f}(\boldsymbol{q}(t), \boldsymbol{u}(t)) \\ \boldsymbol{z}(t) &= \boldsymbol{h}(\boldsymbol{q}(t)) + \boldsymbol{\nu} \end{aligned} \tag{15}$$

where $\boldsymbol{z}(t) \in \mathbb{R}^p$ are the output of the system and $\boldsymbol{\nu} \sim \mathcal{N}(0, \boldsymbol{R}(t))$ is a normally-distributed Gaussian output noise with zero mean and covariance matrix $\boldsymbol{R}(t)$. A suitable metric for quantifying the amount of the acquired information is the aforementioned CG that quantifies the ability of estimating the *current state* $\boldsymbol{q}(t)$ from knowledge of the system outputs $\boldsymbol{z}(\bar{t})$ and inputs $\boldsymbol{u}(\bar{t})$ with $\bar{t} \in [t_0,\, t]$ and hence the amount of information collected through the onboard sensors about $\boldsymbol{q}(t)$. The CG is defined as

$$\boldsymbol{\mathcal{G}}_{\boldsymbol{c}}(t_0, t_f) \triangleq \int_{t_0}^{t_f} \boldsymbol{\Phi}^T(\tau, t_f)\boldsymbol{H}^T(\tau)\boldsymbol{W}_c(\tau)\boldsymbol{H}(\tau)\boldsymbol{\Phi}(\tau, t_f)\,\mathrm{d}\tau \tag{16}$$

where $t_f > t_0$, $\boldsymbol{H}(\tau) = \frac{\partial \boldsymbol{h}(\boldsymbol{q}(\tau))}{\partial \boldsymbol{q}(\tau)}$, and $\boldsymbol{W}_c(\tau) \in \mathbb{R}^{p\times p}$ is a symmetric positive definite weight matrix (a design parameter), that may be used for, e.g., accounting for outputs with different units and different uncertainties. Matrix $\boldsymbol{\Phi}(t, t_f) \in \mathbb{R}^{n\times n}$, also known as *sensitivity matrix*, is defined as $\boldsymbol{\Phi}(t, t_f) = \frac{\partial \boldsymbol{q}(t)}{\partial \boldsymbol{q}_f}$ and obeys the following differential equation with *final* conditions at t_f

$$\dot{\boldsymbol{\Phi}}(t, t_f) = \frac{\partial f(\boldsymbol{q}(t), \boldsymbol{u}(t))}{\partial \boldsymbol{q}(t)}\boldsymbol{\Phi}(t, t_f)\,,\ \boldsymbol{\Phi}(t_f, t_f) = I. \tag{17}$$

The link between the CG and the optimal estimation error covariance matrix $\boldsymbol{P}$ (obtained here with the EKF), reported in [11], turns to be instrumental for a practical implementation of the feedback-feedforward control scheme. More precisely, let us consider the continuous Riccati equation

$$\dot{\boldsymbol{P}}^{-1}(t) = -\boldsymbol{P}^{-1}(t)\boldsymbol{A}(t) - \boldsymbol{A}^T(t)\boldsymbol{P}^{-1}(t) + \boldsymbol{H}^T(t)\boldsymbol{R}^{-1}\boldsymbol{H}(t), \tag{18}$$

with $\boldsymbol{A}(t) = \frac{\partial \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{u})}{\partial \boldsymbol{q}}$, i.e., the state-dependent linearized dynamic matrix of (15) around a nominal trajectory. It is possible to show that the solution of (18) is

$$\boldsymbol{P}^{-1}(t) = \boldsymbol{\Phi}^T(t_0, t)\boldsymbol{P}_0^{-1}\boldsymbol{\Phi}(t_0, t) + \boldsymbol{\mathcal{G}}_{\boldsymbol{c}}(t_0, t) = \boldsymbol{\mathcal{G}}_{\boldsymbol{c}}(-\infty, t)\,. \tag{19}$$

The first term of (19) represents the contribution of the *a priori* information $\boldsymbol{P}^{-1}(t_0) = \boldsymbol{P}_0^{-1}$ but *shifted* at time t by $\boldsymbol{\Phi}(t_0, t)$. The second term is instead the contribution of the information actually collected during the interval $[t_0,\, t]$ and encoded by the CG in (16) with $\boldsymbol{W}_c(t) = \boldsymbol{R}^{-1}(t)$ and $t_f = t$. Hence, $\boldsymbol{\mathcal{G}}_{\boldsymbol{c}}(-\infty, t)$ represents the *current* knowledge, given by the already collected information about the current $\boldsymbol{q}(t)$ in the interval $[t_0,\, t]$ *plus* any other additional a priori information available at time t_0, and it is directly available from the EKF. When planning the *future* maneuvers for $[t,\, t_f]$, one has

$$\boldsymbol{\mathcal{G}}_c(-\infty, t_f) = \Phi^T(t, t_f)\boldsymbol{\mathcal{G}}_{\boldsymbol{c}}(-\infty, t)\boldsymbol{\Phi}(t, t_f) + \boldsymbol{\mathcal{G}}_{\boldsymbol{c}}(t, t_f) \tag{20}$$

where the first term represent bow the current knowledge is projected at the final time t_f by means of the operator $\boldsymbol{\Phi}^T(t, t_f)$ while the second one quantify the information yet to be collected in the interval $[t,\, t_f]$ and correctly represented by CG.

In this work, as in [11], we will use the differential approximation of the smallest eigenvalue of the CG given by

$$\|\boldsymbol{\mathcal{G}}_c(-\infty, t_f)\|_\mu = \sqrt[\mu]{\sum_{i=1}^{n} \lambda_i^\mu(\boldsymbol{\mathcal{G}}_c(-\infty, t_f))} \tag{21}$$

where $\mu << -1$ and $\lambda_i(\boldsymbol{A})$ stands for "the i-th smallest eigenvalue of $\boldsymbol{A}$".

As a consequence, the Information-aware LMPC problem to be solved at runtime for the feedback-feedforward control scheme reported in Fig. 1 reads as follows:

Problem 2 (Information-aware LMPC) *With the same meaning of the parameters of Problem 1, find the optimal sequence, $\forall t \in [t_k, t_{k+L}]$, of the feedforward control components*

$$\begin{aligned}
\boldsymbol{u}^*_{ff} = & \min_{\boldsymbol{u}_{ff} \in S(\Delta)} -\|\boldsymbol{\mathcal{G}}_c(-\infty, t_{k+L})\|_\mu \\
& \textit{s.t.} \\
& 1)\ \dot{\hat{\tilde{\boldsymbol{q}}}}(t) = \boldsymbol{f}(\hat{\tilde{\boldsymbol{q}}}(t)) + \boldsymbol{g}(\hat{\tilde{\boldsymbol{q}}}(t))(\boldsymbol{u}_{ff}(t) + \boldsymbol{u}_{fb}(\hat{\tilde{\boldsymbol{q}}}(t))) \\
& 2)\ \hat{\tilde{\boldsymbol{q}}}(t_k) = \hat{\boldsymbol{q}}(t_k) \\
& 3)\ \underline{\boldsymbol{u}} - \boldsymbol{u}_{fb}(\hat{\tilde{\boldsymbol{q}}}(t)) \leq \boldsymbol{u}_{ff}(t) \leq \bar{\boldsymbol{u}} - \boldsymbol{u}_{fb}(\hat{\tilde{\boldsymbol{q}}}(t)) \\
& 4)\ L_g V(\hat{\boldsymbol{q}}(t_k))\hat{\boldsymbol{u}}_{ff}(t_k) \leq 0
\end{aligned}$$

Notice that, Condition 4) of Problem 2 derives directly from (14), which is the explicit equivalent form of Condition 4) of Problem 1 when $\boldsymbol{u}(t_k) = \boldsymbol{u}_{ff}(t_k) + \boldsymbol{u}_{fb}(t_k)$ is considered.

III. SIMULATION RESULTS

To prove the effectiveness of our approach, we test it on a unicycle vehicle that performs three classical tasks in mobile robotics: path following, point-to-point motion, and trajectory tracking. Moreover, we compare the results applying the proposed Information-aware LMPC (dubbed I-LMPC), i.e., feedback-feedforward controls obtained by the solutions of Problem 2, with: 1) the results obtained by directly applying the feedback $\hat{\boldsymbol{u}}_{fb}(t)$ only (LCL) 2) the results obtained by using the classical LMPC, that is, by applying the solution of Problem 2 where the cost function defined by (21) is replaced by a task-oriented cost function defined as follows:

$$\int_{t_k}^{t_{k+L}} \|\tilde{\boldsymbol{q}}(\tau) - \boldsymbol{q}_{Task}\|_{\boldsymbol{Q}_c} + \|\boldsymbol{u}(\tau) - \boldsymbol{u}_{Task}(\tau)\|_{\boldsymbol{R}_c}\, d\tau \tag{22}$$

where $(\boldsymbol{q}_{Task}, \boldsymbol{u}_{Task})$ are the desired state and control inputs that steer the vehicle once the task is executed.

To this end, we perform 100 simulations for each task and each control approach randomizing on the initial estimated configurations $\hat{\boldsymbol{q}}_0$ that are generated according to the initial estimation covariance matrix $\boldsymbol{P}_0$ related to the initial state estimation. Moreover, we carry out a statistical analysis in terms of estimation error and task error by using a Wilcoxon rank sum test with a significance level of 5%. The task error is obtained by computing the difference between the current configuration of the robot and the one it would assume if the task was correctly executed. We assume a normally distributed Gaussian output noise with zero mean and covariance matrix, $\boldsymbol{R} = 0.3\boldsymbol{I}$ while the actuation/process noise is considered negligible. Finally, the sampling time is $\Delta = 50$ ms for all the tasks, while the prediction horizon L, which is equal to the control horizon, is equal to 30, 15 and 20 time steps for the path following, the point-to-point and the trajectory tracking, respectively. These values are chosen as a trade-off between the computation time and the possibility of correctly executing the task. All the optimization problems are solved using the CasADi tool [23] in Python and adopting the direct single shooting method with the ma57 ipopt solver.

A. Path following

The objective is to determine a Lyapunov based control law such that the vehicle is asymptotically stabilized, w.l.o.g. on the straight line $y = 0$. This goal can be achieved if, at the end, $y \equiv 0$ and $\theta \equiv 0$. Note that for this task, the dynamic of y and θ are not influenced by the one of x. As a consequence, we consider the following reduced kinematic model of the unicycle vehicle,

$$\begin{cases} \dot{y} = \bar{v}\, \sin\theta, \\ \dot{\theta} = \omega, \end{cases}$$

where $\boldsymbol{q} = [y\ \theta]^T$ is the state of the robot and $\boldsymbol{u} = [\bar{v}\ \omega]^T$ the control inputs (with $\bar{v} \neq 0$ assumed constant). Let us hence consider the following Lyapunov candidate function $V(\boldsymbol{q}) = \frac{1}{2}(y^2 + \theta^2)$. By choosing $\omega_{fb}(\boldsymbol{q}) = -y\frac{\sin\theta}{\theta}\bar{v} - K_\theta\theta$ we obtain $\dot{V}(\boldsymbol{q}) = -K_\theta\theta^2 \leq 0$, and hence, by using the Krasowski-Lasalle principle it is easy to show the G.A.S. of the equilibrium.

Since the state of the robot is unknown, the control inputs have to be computed by using $\hat{\boldsymbol{q}}(t)$ instead of $\boldsymbol{q}(t)$. As a consequence, after some algebra and following the step described in II-A, the first two moments in (8) become

$$\begin{aligned}
&\mathrm{E}\left\{\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}})\right\} = -K_\theta\theta^2 + \theta\mathrm{E}\left\{\delta_u(\varepsilon_q)\right\} = -K_\theta\theta^2, \\
&\mathrm{E}\left\{\left(\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}}) - \mathrm{E}\left\{\dot{V}(\boldsymbol{q}, \hat{\boldsymbol{q}})\right\}\right)^2\right\} = \theta^2\boldsymbol{D}\boldsymbol{P}\boldsymbol{D}^T.
\end{aligned}$$

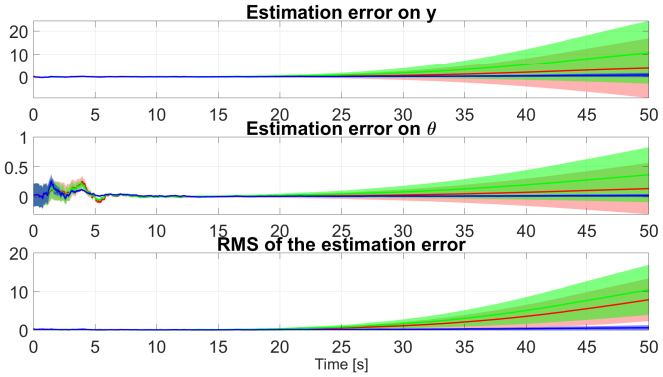
Moreover, the stability constraint of the LMPC (14) for this task becomes

$$\theta\, \omega_{ff} \leq 0. \tag{23}$$

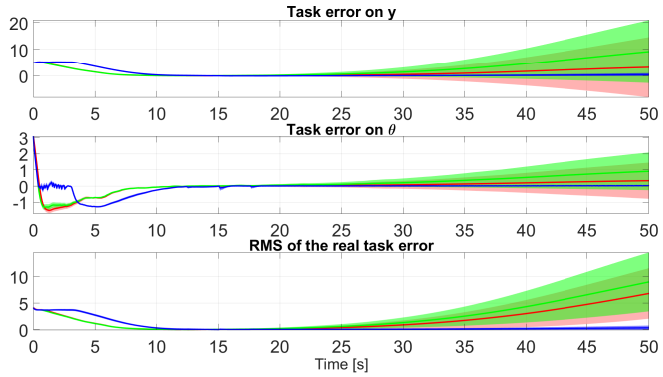
Notice that, for this case, $\boldsymbol{q}_{Task}$ and $\boldsymbol{u}_{Task}$ are equal to zero in (22).

In the simulations, the unicycle is equipped with a sensor that measures the range from the path, i.e., $h(t) = y(t)$. Notice that, the straight line $y = 0$ is an unobservable path with this output (θ is not observable). We choose $T = 18$ sand the initial configuration is $q_0 = [5\text{ m},\ \pi\text{ rad}]^T$ and $\boldsymbol{P}_0 = \mathrm{diag}([0.5^2,\ 0.2^2])$. To conclude, $\bar{v} = 1$ m/s, $K_\theta = 2$ and $-7 \leq \omega_{fb} + \omega_{ff} \leq 7$.

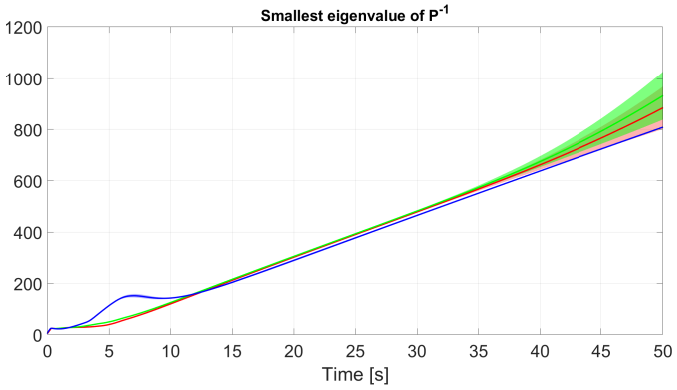
Fig. 2 shows the mean values with their standard deviation of both the estimation errors and the task execution performances. For the LCL and the LMPC, the estimation errors do not converge to zero (see Figure 2(a)) and hence the task is not correctly executed. In addition, for the I-LMPC case, the uncertainty is smaller than the other two cases most of the time. Notice that, as soon as the vehicle approaches the desired path



(a) The value of the mean RMS with standard deviation at the final time are 7.84±5.54 for LCL, 10.4±6.46 for LMPC and 0.57±0.66 for I-LMPC.



(b) The value of the mean RMS with standard deviation at the final time are 6.85±4.76 for LCL, 9.02±5.56 for LMPC and 0.36±0.43 for I-LMPC.



(c) The smallest eigenvalue of the inverse of the estimation covariance matrix of the EKF.

Fig. 2. Statistical results in terms of average value and standard deviations for the path following task. The results obtained for the LCL are plotted in red, for the LMPC in green, for the proposed I-LMPC in blue.

(i.e., approximately after 13 s of simulation, see Figure 2(c)), the smallest eigenvalue of $\boldsymbol{P}^{-1}$ reduces, confirming that the straight line $y = 0$ is an unobservable path. Moreover, the Wilcoxon test confirms that there are statistical differences for all cases, and that our approach provides the most informative trajectories. Fig. 3 shows the trajectories corresponding to one sample out of the 100 performed. Notice that the LCL and LMPC trajectories are similar, and the real robot reaches the path and stays there for a while before diverging indefinitely from it. Instead, by using the I-LMPC solution, the vehicle follows a completely different path and correctly executes the task remains on the path for much longer. The average computation time for each iteration, together with its standard deviation is 20 ± 1 ms for LMPC and 40 ± 2 ms for I-LMPC. Moreover, the task success rate is 6% for LCL, 5% for LMPC and up to 60% for I-LMPC which, as a consequence, outperforms the other methods of comparison. For the sake of space, we only show the trajectories corresponding to the path following task. Please, refer to the accompanying multimedia material for further details and simulations.

B. Point-to-point motion

The objective is the stabilization of an equilibrium point (w.l.o.g. the origin in our case), in the state space of the vehicle whose kinematics is expressed in polar coordinates [23]:

$$\begin{cases} \dot{\rho} = -\rho \cos\alpha w \\ \dot{\phi} = \sin\alpha w \\ \dot{\alpha} = \sin\alpha w - \omega \end{cases}$$

where $\boldsymbol{\xi} = [\rho,\ \phi,\ \alpha]^T$, $w = v/\rho$ where v is the forward velocity and ω is the angular velocity. Let us consider now the following positive definite Lyapunov candidate $V(\boldsymbol{\xi}) = \frac{1}{2}(\lambda_1\rho^2 + \lambda_2\phi^2 + \alpha^2)$ with λ_1 and λ_2 positive parameters, and by choosing

$$\begin{cases} w_{fb}(\boldsymbol{\xi}) = k_1 \cos\alpha, \ \text{with } k_1 > 0 \\ \omega_{fb}(\boldsymbol{\xi}) = k_1 \frac{\sin\alpha}{\alpha} \cos\alpha(\alpha + \lambda_2\phi) + k_2\alpha, \ \text{with } k_2 > 0 \end{cases}$$

we obtain $\dot{V}(\boldsymbol{\xi}) = -\lambda_1 k_1 \rho^2 \cos^2\alpha - k_2\alpha^2$ which is negative semi-definite. Nevertheless, by using the Krasowski-Lasalle principle, we are able to conclude on the G.A.S. of the origin.

Since the state of the robot is unknown, also in this case, the control inputs have to be computed by using $\hat{\boldsymbol{\xi}}(t)$ instead of $\boldsymbol{\xi}(t)$. After some algebra, the first two moments of $\dot{V}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})$ are

$$\begin{aligned} &\mathrm{E}\left\{\dot{V}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})\right\} = -\lambda_1 k_1 \rho^2 \cos^2\alpha - k_2\alpha^2, \\ &\mathrm{E}\left\{\left(\dot{V}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}}) - \mathrm{E}\left\{\dot{V}(\boldsymbol{\xi}, \hat{\boldsymbol{\xi}})\right\}\right)^2\right\} = \bar{\boldsymbol{D}}\boldsymbol{P}\bar{\boldsymbol{D}}^T \end{aligned} \tag{24}$$

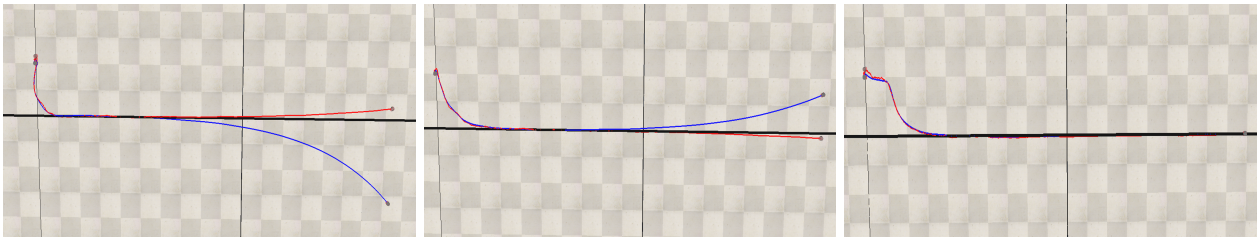
with $\bar{\boldsymbol{D}} = -\begin{bmatrix} -\lambda_1\rho^2\cos\alpha + \lambda_2\phi\sin\alpha + \alpha\sin\alpha & -\alpha \end{bmatrix} \boldsymbol{D}$ and $\boldsymbol{D} = [\boldsymbol{D}_w\ \boldsymbol{D}_\omega]^T$. To conclude, for this task, the stability constraint (14) to be included in the LMPC is:

$$(-\lambda_1\rho^2\cos\alpha + \lambda_2\phi\sin\alpha + \alpha\sin\alpha)w_{ff} - \alpha\omega_{ff} \leq 0\,. \tag{25}$$

Notice that, also for this case, $\boldsymbol{q}_{Task}$ and $\boldsymbol{u}_{Task}$ are equal to zero in (22).

The vehicle is equipped with a sensor that provides range measurements w.r.t. two markers located in $(0, \pm 2)$. The initial robot configuration is $\bar{q}_0 = [5.1\text{ m},\ -2.94\text{ rad},\ 0.2\text{ rad}]$ and $\boldsymbol{P}_0 = \mathrm{diag}([0.7^2,\ 0.25^2,\ 0.3^2])$. Moreover, $T = 6$ s, $k_1 = k_2 = 1.8$ and $\lambda_1 = 1$ and $\lambda_2 = 0.5$. Finally, $0 \leq w_{fb} + w_{ff} \leq 2$ and $-3 \leq \omega_{fb} + \omega_{ff} \leq 3$.

In Fig. 4, the estimation errors and the task execution performances are shown. In Fig. 4(a), the estimation errors for the LMPC do not converge to zero even if it does not diverge as for the path following case. Moreover, from Fig. 4(b) the task



(a) LCL. (b) LMPC. (c) I-LMPC.

Fig. 3. Robot real/estimated trajectories on the plane of motion for the path following task. The trajectories correspond to one of the 100 simulations. The real robot trajectory from the real robot configuration $\boldsymbol{q}(t_0) = [0,\ 5,\ 3.14]^T$ is plotted in blue, the estimated robot trajectory from the estimated configuration $\hat{\boldsymbol{q}}_0 = [0,\ 5.64,\ 3.61]^T$ is in red and the desired straight line path in black

error for the state variable ϕ converges to zero only if the I-LMPC is used. As a consequence, the use of an active sensing control strategy as a feedforward component guarantees the best execution of the task. Fig. 4(c) shows the trend of the smallest eigenvalue of $\boldsymbol{P}^{-1}$ where the estimation uncertainty is consistently reduced with the proposed I-LMPC. Finally, the Wilcoxon test shows statistical differences, confirming that the use of the proposed I-LMPC allows guaranteeing the best performance in terms of task execution. The average computation time for iteration with standard deviation is 8 ± 2 ms for LMPC and 17 ± 9 ms for I-LMPC.

C. Trajectory tracking

The objective is to design a Lyapunov-based control law that allow the unicycle to track a leader unicycle that starting from an initial configuration $q_l = [x_l,\ y_l,\ \theta_l]^T$ moves on a desired trajectory with linear and angular velocity, v_l and w_l, respectively.

Let us hence consider the kinematic of the tracking error $\boldsymbol{e} = [x - x_l, y - y_l, \theta - \theta_l]^T$,

$$\dot{\boldsymbol{e}} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} v + e_2\omega - v_l \cos e_3 \\ -e_1\omega + v_l \sin e_3 \\ \omega - \omega_l \end{bmatrix} \tag{26}$$

and the following positive definite Lyapunov candidate $V(\boldsymbol{e}) = \frac{1}{2}(e_1^2 + e_2^2) + K(1 - \cos e_3)$ with $K > 0$. By choosing the following control law

$$\begin{cases} v_{fb}(\boldsymbol{e}) = v_l \cos e_3 - K\lambda_1 e_1 \\ \omega_{fb}(\boldsymbol{e}) = \omega_l - \frac{e_2}{K} v_l - \lambda_2 \sin e_3 \end{cases}$$

with $\lambda_1, \lambda_2, K > 0$, we obtain $\dot{V}(\boldsymbol{e}) = -\lambda_1 e_1^2 - K\lambda_2 \sin^2 e_3 \leq 0$ and then by using the Krasowski-Lasalle principle, the G.A.S. of the equilibrium $\boldsymbol{e} \equiv 0$ can be demonstrated.

However, again, since the control law is computed by using the estimated state, the first two moments of $\dot{V}(\boldsymbol{e}, \hat{\boldsymbol{e}})$ are

$$\begin{aligned} &\mathrm{E}\left\{\dot{V}(\boldsymbol{e}, \hat{\boldsymbol{e}})\right\} = -\lambda_1 e_1^2 - \lambda_2 K \sin^2 e_3, \\ &\mathrm{E}\left\{\left(\dot{V}(\boldsymbol{e}, \hat{\boldsymbol{e}}) - \mathrm{E}\left\{\dot{V}(\boldsymbol{e}, \hat{\boldsymbol{e}})\right\}\right)^2\right\} \\ &= \left(-\begin{bmatrix} e_1 & K \sin e_3 \end{bmatrix}\right) \boldsymbol{D}\boldsymbol{P}\boldsymbol{D}^T \left(-\begin{bmatrix} e_1 & K \sin e_3 \end{bmatrix}\right)^T \end{aligned} \tag{27}$$

with $\boldsymbol{D} = [\boldsymbol{D}_v\, \boldsymbol{D}_\omega]^T$. For this task, the Lyapunov constraint (14) becomes:

$$e_1 v_{fb} + K \sin e_3\, \omega_{fb} \leq 0\,. \tag{28}$$

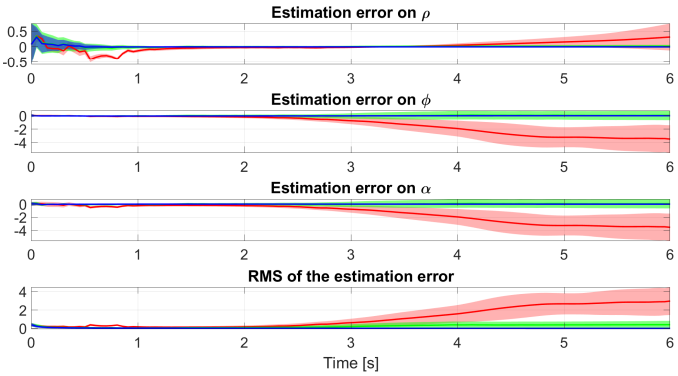
Notice that, for this case, $\boldsymbol{q}_{Task} = q_l$ and $\boldsymbol{u}_{Task} = [v_l\,, \omega_l]^T$ in (22).

The vehicle exploits the range measurements w.r.t. two markers located in $(0, 2)$ and $(0, 0)$. The initial configuration of the leader unicycle is $\boldsymbol{q}_{l0} = [-1\text{m},\ 0\text{m},\ 0\text{rad}]^T$, and moves with $v_l = 0.5$ m/s and $\omega_l = 0.5$ rad/s. The initial configuration of the real robot is $\boldsymbol{q}_0 = [-1\text{m},\ 1\text{m},\ 0\text{rad}]^T$ and $\boldsymbol{P}_0 = \mathrm{diag}([0.2^2,\ 0.2^2,\ 0.1^2])$. Moreover, $T = 30$ s, $K = 0.8$ and $\lambda_1 = \lambda_2 = 0.6$. Finally, $0 \leq v_{fb} + v \leq 3$ and $-3 \leq \omega_{fb} + \omega \leq 3$. In Fig. 5(a), the estimation errors converge for all the cases. Notice that, even if the estimation errors are comparable, by using the LMPC the real robot performs the task quite better reaching the leader first with the smallest task error. However, in Fig. 5(c) the estimation uncertainty for the I-LMPC is the smallest one as the smallest eigenvalue of $\boldsymbol{P}^{-1}$ is the largest for the most of the simulation time. Coming to the uncertainties evaluation, the Wilcoxon test shows that the trajectories obtained with our methodology are statistically more informative than the others, and hence the I-LMPC can be considered more reliable than both the LCL and the LMPC.

For this test case the average computation time for iteration with standard deviation is 6 ± 1 ms for LMPC and 9 ± 1 ms for I-LMPC.

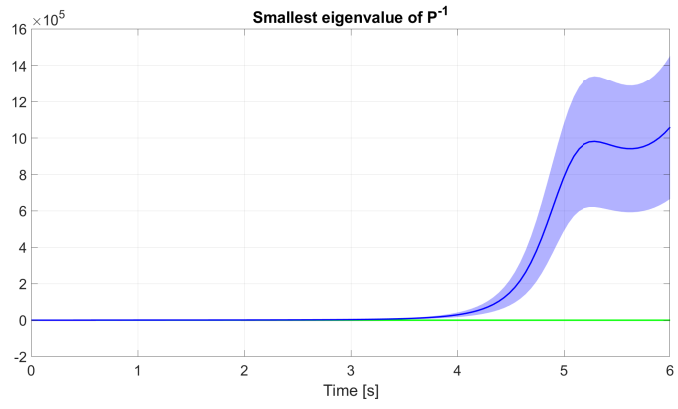
IV. CONCLUSIONS AND FUTURE WORKS

This paper proposed a feedback-feedforward Information-aware LMPC control scheme that combines the benefits of an online active sensing control strategy and a Lyapunov-based control strategy. The smallest eigenvalue of the Constructability Gramian was adopted to quantify the richness of the acquired information. We showed in simulations on a unicycle vehicle that our methodology guarantees better estimation performance, and hence a better task execution in general. It is worthwhile to note that the I-LMPC generates predicted trajectories that are apparently less task-oriented than those obtained for the LCL and LMPC. This is because the Lyapunov constraint is imposed for the next control step, while the remaining steps maximize only the information acquired along the path. Hence, by considering the Lyapunov constraint applied for longer time intervals, we can ensure better stability



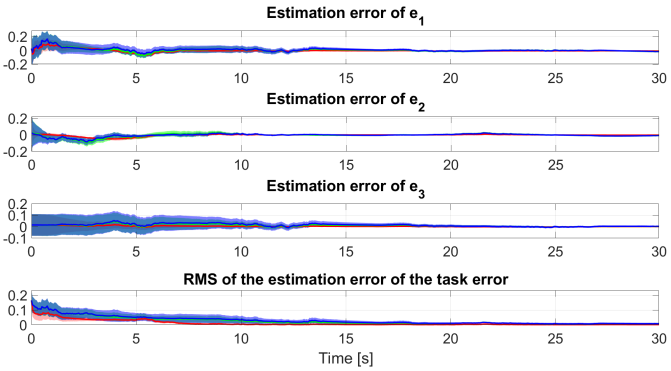
(a) The value of the mean RMS with standard deviation at the final time are 2.98±1.52 for LCL, 0.4±0.42 for LMPC and 0.001±0.001 for I-LMPC.

(b) The value of the mean RMS with standard deviation at the final time are 2.91±0.95 for LCL, 0.84±0.43 for LMPC and 0±0 for I-LMPC.

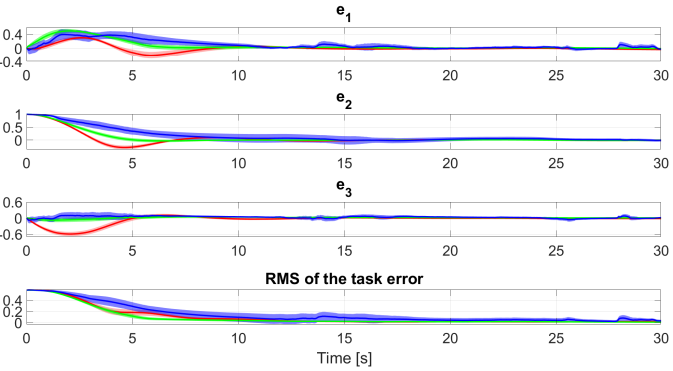


(c) The smallest eigenvalue of the inverse of the estimation covariance matrix of the EKF.

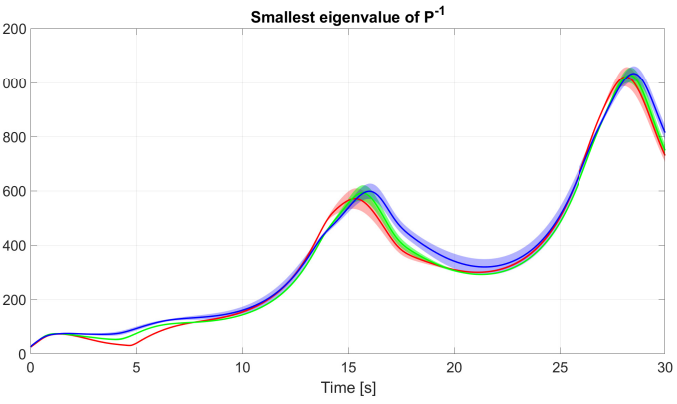
Fig. 4. Statistical results in terms of average value and standard deviations for the point-to-point task. The results obtained for the LCL are plotted in red, for the LMPC in green, for the proposed I-LMPC in blue.



(a) The value of the mean RMS with standard deviation at the final time are 0.001±0.00 for LCL, 0.01±0.003 for LMPC and 0.01±0.003 for I-LMPC.



(b) The value of the mean RMS with standard deviation at the final time are 0.021±0.01 for LCL, 0.01±0.003 for LMPC and 0.03±0.02 for I-LMPC.



(c) The smallest eigenvalue of the inverse of the estimation covariance matrix of the EKF.

Fig. 5. Statistical results in terms of average value and standard deviations for the trajectory tracking task. Statistical results in terms of average value and standard deviations for the path following. The results obtained for the LCL are plotted in red, for the LMPC in green, for the proposed I-LMPC in blue.

properties and have more task-oriented trajectories, a promising direction that will be explored in next investigations. Our methodology also presents oscillations in the control inputs, due to the negative effects of the measurement noise, and some sharp maneuvers, due to the persistent requirement of information maximization. The latter aspect will be solved in future works by considering an adaptive cost index that disconnects the feedforward part when the estimation uncertainty is below a desired threshold. Moreover, additional future works will deal with the extension of our methodology to more complex Lyapunov control techniques, as e.g., adaptive control as well as more complex robots (e.g., quadrotors) in real time on a real experiment, also considering the degrading effects of actuation noise. We also plan to use our method as a risk-aware control scheme where the feedforward component maximize the information on the surrounding risks while the feedback component is used for the task execution in a risky environment.

REFERENCES

[1] A. A. Faisal and D. M. Wolpert, "Near optimal combination of sensory and motor uncertainty in time during a naturalistic perception-action task," *Journal of Neurophysiology*, vol. 101, no. 4, pp. 1901–1912, 2009.

[2] S.-H. Yeo, D. Franklin, and D. Wolpert, "When optimal feedback control is not enough: Feedforward strategies are required for optimal control with active sensing," *PLoS computational biology*, vol. 2, no. 3, p. e1005190, 2016.

[3] J. J. Gibson, *The senses considered as perceptual systems*. Boston, MA: Houghton Mifflin, 1966.

[4] S. M. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[5] R. Bajcsy, Y. Aloimonos, and J. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, no. 2, pp. 177–196, 2018.

[6] A. T. Taylor, T. A. Berrueta, and T. D. Murphey, "Active learning in robotics: A review of control principles," *Mechatronics*, vol. 77, p. 102576, 2021.

[7] X. Ren, J. Luo, E. Solowjow, J. A. Ojea, A. Gupta, A. Tamar, and P. Abbeel, "Domain randomization for active pose estimation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7228–7234.

[8] T. Greigarn, M. S. Branicky, and M. C. Çavuşoglu, "Task-oriented active sensing via action entropy minimization," *IEEE Access*, vol. 7, pp. 135 413–135 426, 2019.

[9] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," in *2019 American Control Conference (ACC)*, 2019, pp. 3936–3943.

[10] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1–8.

[11] P. Salaris, M. Cognetti, R. Spica, and P. Robuffo Giordano, "Online optimal perception-aware trajectory generation," *IEEE Transactions on Robotics*, vol. 35, no. 6, pp. 1307–1322, 2019.

[12] M. Cognetti, P. Salaris, and P. Robuffo Giordano, "Optimal active sensing with process and measurement noise," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2118–2125.

[13] O. Napolitano, D. Fontanelli, L. Pallottino, and P. Salaris, "Gramian-based optimal active sensing control under intermittent measurements," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[14] M. Cognetti, M. Aggravi, C. Pacchierotti, P. Salaris, and P. Giordano Robuffo, "Perception-aware human-assisted navigation of mobile robots on persistent trajectories," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4711–4718, 2020.

[15] P. Bernard, N. Mimmo, and L. Marconi, "On the semi-global stability of an ek-like filter," *IEEE Control Systems Letters*, vol. 5, no. 5, pp. 1771–1776, 2021.

[16] V. Magnago, M. Andreetto, S. Divan, D. Fontanelli, and L. Palopoli, "Ruling the Control Authority of a Service Robot based on Information Precision," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia: IEEE, May 2018, pp. 7204–7210.

[17] P. Christofides, J. Liu, and M. Peña, *Networked and Distributed Predictive Control: Methods and Nonlinear Process Network Applications*. Springer, 2011.

[18] J. Liu, D. Muñoz de la Peña, P. D. Christofides, and J. F. Davis, "Lyapunov-based model predictive control of nonlinear systems subject to time-varying measurement delays," *International Journal of Adaptive Control and Signal Processing*, vol. 23, no. 8, pp. 788–807, 2009.

[19] D. Munoz de la Pena and P. D. Christofides, "Lyapunov-based model predictive control of nonlinear systems subject to data losses," *IEEE Transactions on Automatic Control*, vol. 53, no. 9, pp. 2076–2089, 2008.

[20] C. Shen, Y. Shi, and B. Buckham, "Trajectory tracking control of an autonomous underwater vehicle using lyapunov-based model predictive control," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5796–5805, 2018.

[21] R. Grandia, A. Taylor, A. Singletary, M. Hutter, and A. Ames, "Nonlinear model predictive control of robotic systems with control lyapunov functions," in *Robotics Science and Systems*, 2020.

[22] E. A. Buehler, J. A. Paulson, and A. Mesbah, "Lyapunov-based stochastic nonlinear model predictive control: Shaping the state probability distribution functions," in *2016 American Control Conference (ACC)*, 2016, pp. 5389–5394.

[23] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle like vehicles via lyapunov techniques," *IEEE Robotics Automation Magazine*, vol. 2, no. 1, pp. 27–35, 1995.